

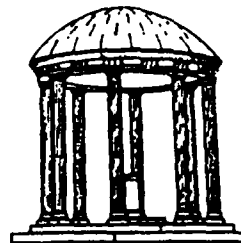
AD-A208 087

Walkthrough - Exploring Virtual Worlds  
Second Annual Report

TR89-012  
February 1989

John M. Airey  
F. P. Brooks

The University of North Carolina at Chapel Hill  
Department of Computer Science  
CB#3175, Sitterson Hall  
Chapel Hill, NC 27599-3175



This work was supported by NSF Grant#CCR-8609588 and  
ONR Grant#N00014-86-K-0680.  
UNC is an Equal Opportunity/Affirmative Action Institution.

# Walkthrough—Exploring Virtual Buildings

Second Annual Technical Progress Report  
to the National Science Foundation  
Computer Science and Engineering  
and the Office of Naval Research

John M. Airey, F. P. Brooks  
February 16, 1989

## Part I - Prose Report

### Introduction

A *virtual world system* is a computer graphics system allowing natural real-time interaction with a 3-dimensional model world. Walkthrough is a virtual world system for architects and their clients to explore buildings before they are built, so as to iteratively refine their specifications.

Much of our virtual-world research at Chapel Hill has involved molecules or human anatomy. In neither of these applications does one know exactly how the virtual world *should* seem to the explorer. In Walkthrough, on the other hand, the researcher can easily tell how closely the virtual world matches the real in the ways that matter.

Any virtual-world research project must continually improve in four aspects, shown in the figure. The videotape, which constitutes Part II of this report, shows where we are today, and gives short previews of some of the advances not yet integrated into the system.

The figure also serves as the outline of this prose report. Each of the items shown in ordinary type in the figure represents accomplishments since the first annual progress report in the summer of 1988; each shown in italics indicates things we plan, most of which we will achieve in 1989.

### Update Rate

Natural motion is *the* crucial factor for realistic interaction. This is achieved through faster image display rates. Display speed also limits other effects important to image realism, namely, realistic shading and texturing, level of detail, and stereoscopic viewing. We have developed preprocessing software algorithms that accelerate image display on our current hardware. These algorithms will probably remain important through several future generations of image generation hardware as we can envision image improvements that will soak up all excess display power.

The system currently in use is the Pixel-Planes 4 system which can display close to 30,000 gouraud shaded polygons a second. Using a model with 8,000 polygons gave us about 4 updates per second. We used a program developed by John Airey to automatically partition the model which allowed update rates of 15, but which presented the viewer with some views that were not correct, such as at the boundary between partitions. This algorithm has been extended by Airey to compute the subset of polygons necessary for a correct image for each partition. Thus the viewer sees a correct image everywhere in the model.

Because the algorithm takes a long time to run, we have only applied it to an old version of the building model, because the newer, more accurate model is still changing frequently. This older version has incorrect colors which you may notice in the video. The automatic partitioning algorithm splits the model into 269 partitions or *cells*. The average number of polygons which must be displayed to get a correct image for any cell is about 237. This means that we have obtained a 30-fold average speedup. The maximum number of polygons which must be displayed for any cell is about 2200, which means we have a worst case speedup of 3.25.

# Walkthrough Research Progress and Plan

## Update Rate

Automatic partitioning with automatic visibility set calculations yields correct images and an average 30 fold speed over naive use of Pixel-planes to get > 15 updates/sec.

*Integration of this algorithm with the radiosity lighting model*

## Interface

Flexible Interface Testbed  
Joysticks- free and restricted  
Big Screen  
Treadmill  
Head-Mounted Display  
Map vs. Scene Navigation

*Interface comparison study.  
Dynamic Objects: lights, doors, etc.*

Funding by  
NSF  
ONR

Accomplished  
*Planned*

Walkthrough  
Progress since  
1988 report

## Radiosity Lighting Model

Seven-fold speed improvement over published algorithm through the use of adaptive sampling methods.

Development of an algorithm to filter finite-elements for smoother image

Individual real-time control of lights.

*improve image quality / polygon ratio*

## Model Building

AutoCad  
Model Translator  
Model Massaging  
Furniture  
Lighting specifications

*Real Collaborator*

We expect to test the algorithm on other data sets, in particular the Orange United Methodist Church. We plan to integrate this algorithm with our advances in the use of the radiosity lighting model which greatly improves the realism of the image but makes drastic demands on our update rate. At present we are limited to using radiosity on small data sets or moving very slowly through larger radiosity data sets.

### Radiosity Lighting Model

The primary attack for increasing the realism in the Walkthrough project this year has been the use of the radiosity lighting model, pioneered by Donald Greenberg et al. at Cornell. This model is especially applicable to interiors lit with artificial lighting. It models the physics of radiation transfer in enclosed areas with methods akin to finite element analysis. All surfaces are diced into smaller subsurfaces. A program calculates the amount of energy that each subsurface receives and radiates per unit time. This value is then used to determine the color of each subsurface. The results are interpolated during display to give a smooth variation in color, resulting in the shaded images seen in the videotape.

At the time of the last report, the best radiosity algorithm known cost time and disk space proportional to the square of the number of subsurfaces in the model. This effectively precluded its use on something as large as a building. In August 1988, the Cornell group published a new technique which had linear space requirements. John Airey has implemented this approach and has further improved its time performance four-fold through the use of adaptive sampling methods. Further adaptive methods developed with Ming Ouh-Young have been shown to yield a seven-fold speedup. We plan to incorporate these new methods in our main model this year.

Radiosity researchers have typically used specially designed data sets which simplify several problems, such as the interpolation step. A significant contribution by the Walkthrough project has been our use of data sets designed with a commercially available CAD program, AutoCad. The real-life physics nature of the radiosity lighting model calculation has exposed a whole set of problems arising in real models, as opposed to "nice" artificial ones: interpenetrating polygons, coincident polygons, and nonorientable surfaces.

John Airey developed an interpolation and filter algorithm to remove shading discontinuities from data sets that were not designed with the needs of radiosity display in mind. John Airey, John Rohlf, Clement Cheung, and Randy Brown have built other tools to deal with other model inconsistencies.

The radiosity lighting calculation is a linear system and thus we can apply the principle of superposition to separate the effects of individual lights in a model. This allows us to *interactively* modulate the intensity of individual lights in the model. We have tested this on small data sets such as the room with two lights seen in the videotape. Penny Rheingans, Randy Brown and John Rohlf have modelled the larger lighting systems installed in the lobby of Sitterson Hall and The Orange Baptist Church. We hope to produce images which accurately duplicate these lighting systems. As the video was in production, we managed to run radiosity on the Sitterson Hall lobby. Seventeen independent sets of light sources were modelled, including daylight. The model consists of 25,000 surface patches. This makes it difficult to operate in real-time. The update rate is about 1 per 10 seconds. We will increase this through the use of micro-coded display routines rather than display routines written in C. In the title and credit sequences you can see the office lights come on, followed by the indirect lighting in the recessed ceiling turning off.

Other plans are to improve the image-quality-per-polygon ratio of radiosity images. It may be possible to dice the surfaces intelligently so as to maximize the effect on the viewer without increasing the number of polygons that must be displayed.

Another means to improve realism is the use of texturing. Rohlf has successfully implemented the texturing algorithm proposed for Pixel Planes. There is not enough memory available at each pixel to allow this algorithm to be successfully integrated with the Walkthrough display system, but we expect to experiment with texturing more in the coming year.

## Model Building

The first building model was built by Dana Smith using the Sitterson Hall blueprints, the Caesar chip-layout software from UC-Berkeley and several special purpose C programs. The specialized nature of this modelling system makes it difficult for anyone other than Dana to effectively modify this model. However, it is testimony to the difficulty of modelling that we continue to use this model to test our display acceleration algorithms.

We have adopted the AutoCad program for all future modelling efforts. It handles several problems very well, the most important being its function as a database and file manager. Any member of the team can contribute to the modelling effort. More important is the ability to import AutoCad databases from architects. On the other hand, AutoCad was not designed to make models suitable for photorealistic rendering, but rather for blueprint-style drawings. This has forced us to spend considerable modelling resources developing tools to correct these models, but we judge the benefits of the use of AutoCad to outweigh these disadvantages. One of these tools is demonstrated in the video-tape. A re-tessellation and filtering algorithm was developed by Airey to allow Pixel-Planes to display a smooth image from discrete patches produced by the radiosity calculation.

A major effort in modelling has been to encode lighting system information into the AutoCad model and augment our file translators to interpret this information properly. A model designed for a virtual world is quite complex; one wants a specialized language to encode the characteristics of that world. The translator developed by Penny Rheingans and John Airey, based on the Unix compiler tools, Lex and Yacc, has proved to be easy to extend and maintain.

Our plans for the future are to work with a real collaborator, an architect with a building in AutoCad format, to help us identify important areas to work on.

## Man - Machine Interface

Curtis Hill has spent months re-engineering the software used for the interface to the display system, to make it easy to maintain and flexible. The evaluation of interfaces is necessarily an empirical process. We plan to use this newly engineered system to evaluate interface performance through psychological studies.

For example, the joysticks device, two three-degree-of-freedom joysticks with a slider, has proved to be very useful to those who use the system often, but they require practice and do not properly give a sense of space to the user. The steerable treadmill system on the other hand is very appropriate for those who use the system less often, such as clients. It has a very low learning curve and gives a good sense of space to the user. However, it is inappropriate for the software developers because it takes longer to get to a certain desired vantage point. The head-mounted display system appears to offer the greatest rewards, but the technology is still primitive.

Because the developers use the system everyday, it is difficult to judge interface systems objectively. A controlled experiment that tested learning curves and ease of use would be an important research contribution.

## Part II - Walkthrough Videotape 1989

The second, and more vivid, part of our 1989 Technical Progress Report consists of a VHS videotape that illustrates the progress. It is enclosed herewith. For those without viewing facilities, we include the script.

### Script

*Title over Sitterson Lobby displayed using radiosity lighting model. Watch the office lights on the lower level.*

UNC's Walkthrough system helps architects and their clients by letting them explore buildings while still on paper. The research purpose is to study computer graphics systems for virtual worlds.

*Four way diagram of research expansion*

Virtual world systems research should expand in four directions.

First the update rate must be sufficient for a realistic motion effect.

Any excess computation power may be used to improve image realism. The radiosity lighting model is a step in this direction.

Encoding the information used to create virtual worlds requires sophisticated modelling systems.

Intuitive user interfaces must be created and tested on virtual world systems.

*Walk down hall without the benefit of display acceleration preprocessing algorithm.*

Naive use of Pixel-Planes 4 on a simple model of the building yields a 4-update-per-second display rate. This model has 7125 polygons.

*Walk down the hall and into the lobby using Airey's display acceleration algorithm.*

We have developed a preprocessing algorithm that yields an average 30-fold speedup and worst case 3-fold speedup on this simple model.

The model space is recursively subdivided, by planes perpendicular to the coordinate axes, into smaller volumes or cells. This produces a tree data structure, with cells at the leaf nodes and splitting planes at the interior nodes. The tree used in this example has 269 cells.

The subset of the model that is visible from all viewpoints within a cell is then computed for each of the cells. This required several days of Sun 4 computer time.

At display time, the tree data-structure is searched each frame, using the viewer's position as the key, to find the subset of the model that must be displayed.

We have modified this display program to explicitly show the boundaries between cells as black polygons.

Another debugging option allows us to continue to use the display list for a cell even while we are outside it. This feature can be used to show how polygons that are hidden to an observer in the cell are not displayed. The Carolina-blue color is the background color. Polygons not visible from the northeast entrance are not displayed.

### *Display room with furniture*

With the speed gained from using the preprocessing algorithm we can add furniture and apply a more complicated lighting model.

The radiosity lighting model can produce stunning images but greatly slows the update rate.

### *Display room with furniture cut into radiosity patches.*

An energy transfer calculation computes the amount of energy leaving each patch per unit time in a preprocessing step. This model required only a few minutes of preprocessing time.

### *Display room with furniture with interpolated patches.*

The energy values of the patches are then transferred to polygon vertices with a preprocessing filter. At display time, Pixel Planes linearly interpolates the vertex colors to give smooth shading. The radiosity calculation is a linear system and this allows us to apply the principle of superposition. The contribution of each light may be computed independently in the preprocessing step. A weighted sum computed at display time allows us to modulate the lights in real-time.

### *Display church Fellowship Hall.*

We have improved the performance of the radiosity calculation so that it can be run on large data sets. We have also developed several utilities that make it feasible to run radiosity on data not explicitly designed for the radiosity lighting model.

### *Display wall of church hall, raw patches.*

We have developed a program to re-tessellate and filter patch data obtained from AutoCad data, which would otherwise exhibit shading discontinuities. The raw patch data appears here.

### *Display wall of church hall, averaged with naive algorithm.*

The naive algorithm performs poorly on this real example.

### *Display wall of church hall, re-tessellated and averaged with Airey's algorithm.*

Our re-tessellation and filter algorithm removes the shading discontinuities.

### *AutoCad display.*

Our primary advances in the modelling area since the last report have been to use AutoCad to encode lighting system information in the model.

### *Treadmill and big screen display.*

We have re-engineered the user interface system to emphasize flexibility. We hope to use this feature to do controlled studies of interface performance. The devices include the treadmill with big screen display shown here, joysticks, a head-mounted display system and a bicycle.

### *Lobby, radiositized.*

### *Screen credits.*

